NASA/CR-1998-206910
ICASE Report No. 98-6

Multigrid Strategies for Viscous Flow Solvers on Anisotropic Unstructured Meshes

Dimitri J. Mavriplis
ICASE

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, VA

Operated by Universities Space Research Association

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contracts NAS1-19480 & NAS1-97046

January 1998

MULTIGRID STRATEGIES FOR VISCOUS FLOW SOLVERS ON ANISOTROPIC UNSTRUCTURED MESHES

DIMITRI J. MAVRIPLIS *

Abstract. Unstructured multigrid techniques for relieving the stiffness associated with high-Reynolds number viscous flow simulations on extremely stretched grids are investigated. One approach consists of employing a semi-coarsening or directional-coarsening technique, based on the directions of strong coupling within the mesh, in order to construct more optimal coarse grid levels. An alternate approach is developed which employs directional implicit smoothing with regular fully coarsened multigrid levels. The directional implicit smoothing is obtained by constructing implicit lines in the unstructured mesh based on the directions of strong coupling. Both approaches yield large increases in convergence rates over the traditional explicit full-coarsening multigrid algorithm. However, maximum benefits are achieved by combining the two approaches in a coupled manner into a single algorithm. An order of magnitude increase in convergence rate over the traditional explicit full-coarsening algorithm is demonstrated, and convergence rates for high-Reynolds number viscous flows which are independent of the grid aspect ratio are obtained. Further acceleration is provided by incorporating low-Mach-number preconditioning techniques, and a Newton-GMRES strategy which employs the multigrid scheme as a preconditioner. The compounding effects of these various techniques on speed of convergence is documented through several example test cases.

Key words. multigrid, anisotropic, Navier-Stokes

Subject classification. Applied and Numerical Mathematics

1. Introduction. Multigrid methods have proven to be very effective techniques for accelerating convergence to steady state of both elliptic and hyperbolic problems. For simple elliptic problems, such as a Poisson equation, convergence rates of 0.1 are achievable, meaning that for each multigrid cycle, the numerical error can be reduced by one order of magnitude.

For hyperbolic problems, such as the Euler equations in computational fluid dynamics, the best rate that theoretically can be achieved for a second order discretization is 0.75, according to the analysis discussed by Mulder [24]. Indeed, many structured as well as unstructured Euler solvers achieve convergence rates close to 0.75 [30, 36, 15, 25, 26]. However, for high-Reynolds number viscous flow solutions, multigrid Navier-Stokes solvers generally result in convergence rates which are an order of magnitude or more slower than those obtained for inviscid flows. The main reason for this breakdown in efficiency of the multigrid algorithm is the use of highly stretched anisotropic meshes which are required to efficiently resolve boundary layer and wake regions in viscous flows. Indeed, the higher the Reynolds number, the more grid stretching is required, and the worse the convergence rate becomes. This poses particular difficulties for simulating flight Reynolds number flows for large aircraft, where the required meshes may contain stretching ratios in excess of 100,000 to 1.

The classic multigrid remedy for this problem is to resort to semi-coarsening, or to employ smoothers which are implicit in the direction normal to the stretching [3]. The idea of semi-coarsening is to coarsen the

*ICASE, Mail Stop 403, NASA Langley Research Center, Hampton, Virginia 23681-0001, dimitri@icase.edu. This research was supported by the National Aeronautics and Space Administration under NASA Contract Nos. NAS1-19480 and NAS1-97046 while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681-0001.

mesh only in the direction normal to the grid stretching, rather than in all coordinate directions simultaneously. This idea was used by Mulder [24, 23] to overcome the stiffness associated with the grid alignment phenomenon for an upwind scheme on non-stretched structured meshes. Since different regions of the flow field may contain anisotropies in differing directions, a complete sequence of grids, each coarsened in a single coordinate direction is generally required. Radespiel and Swanson [29] employed semi-coarsening to alleviate the stiffness due to stretched meshes for viscous flow calculations. More recently, Allmaras [1] has shown how the use of preconditioners coupled with semi-coarsening can help alleviate grid stretching induced stiffness. Pierce and Giles [26] have demonstrated improved convergence rates for turbulent Navier-Stokes flows using diagonal preconditioning coupled with a J-coarsening technique on structured grids, where the grid is only coarsened in the J-coordinate direction, which is normal to the boundary layer.

Semi-coarsening techniques can be generalized to unstructured meshes as directional coarsening methods. Graph algorithms can be constructed to remove mesh vertices based on the local degree and direction of anisotropy in either the grid or the discretized equations. This is achieved by basing point removal decisions on the values of the discrete stencil coefficients. This is the basis for algebraic multigrid methods [34], which operate on sparse matrices directly, rather than on geometric meshes. These techniques are more general than those available for structured meshes, since they can deal with multiple regions of anisotropies in conflicting directions. They offer the possibility of constructing algorithms which attempt to generate the "optimal" coarse grid for the problem at hand. Morano et al. [21] have demonstrated how such techniques can produce almost identical convergence rates for a Poisson equation on an isotropic cartesian mesh, and a highly stretched unstructured mesh. More recently, Francescatto [6] has demonstrated convergence improvements for the Navier-Stokes equations using directional coarsening multigrid.

One of the drawbacks of semi- or directional-coarsening techniques is that they result in coarse grids of higher complexity. While a full-coarsening approach reduces grid complexity between successively coarser levels by a factor of 4 in 2D, and 8 in 3D, semi-coarsening techniques only achieve a grid complexity reduction of 2, in both 2D and 3D. This increases the cost of a multigrid V-cycle, and makes the use of W-cycles impractical. Perhaps more importantly for unstructured mesh calculations, the amount of memory required to store the coarse levels is dramatically increased, particularly in 3D. Raw [31] advocates the use of directional coarsening, but at a fixed coarsening rate of 10 to 1, in order to reduce overheads. This generally results in the removal of multiple neighboring points in the coarsening process, and thus requires a stronger smoother than a simple explicit scheme. An alternative to semi-coarsening is to use a line solver in the direction normal to the grid stretching coupled with a regular full coarsening multigrid algorithm, at least for structured grid problems [3].

In the following sections, we examine the benefits obtained through the use of directional coarsening and implicit line solvers, and combine the two approaches to construct an efficient Reynolds averaged Navier-Stokes solver for very highly stretched meshes. The resulting algorithm is then augmented by a preconditioning technique and a Krylov method.

2. Base Solver . The Reynolds averaged Navier-Stokes equations are discretized by a finite-volume technique on meshes of mixed triangular and quadrilateral elements. A sample mesh is depicted in Figure 2.1. Isotropic triangular elements are employed in regions of inviscid flow, and stretched quadrilateral elements are used in the boundary layer and wake regions. All elements of the grid are handled by a single unifying edge-based data-structure in the flow solver [19]. Triangular elements could easily be employed in the boundary layer regions simply by splitting each quadrilateral element into two triangular elements.

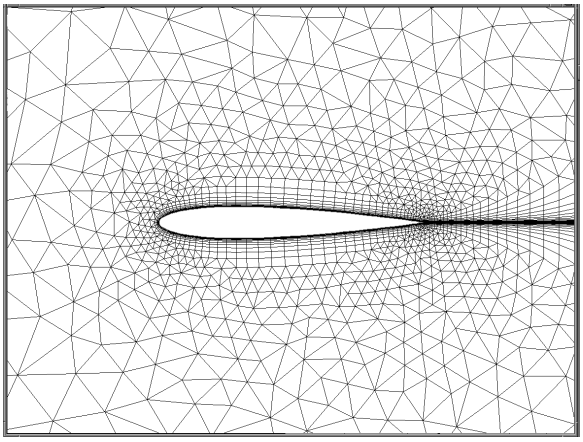

FIG. 2.1. *Mixed element grid used for viscous flow calculations about NACA 0012 airfoil; Number of vertices = 4880*

As shown in Figure 2.2, the resulting control-volumes for quadrilateral elements produce stencils with strong coupling in the direction normal to the grid stretching (i.e. large control volume faces) and weak coupling in the direction of stretching. When triangular elements are employed in regions of high mesh stretching, the stencils are complicated by the presence of diagonal connections, and do not decouple as simply in the normal and stretching directions as for quadrilateral elements. Therefore, the use of quadrilateral elements in regions of high mesh stretching is central to the solution algorithms described in this paper. Alternatively, one could chose to retain triangular elements throughout the entire mesh, and employ different types of dual control-volumes, such as a containment dual rather than median-dual based control-volume [2].

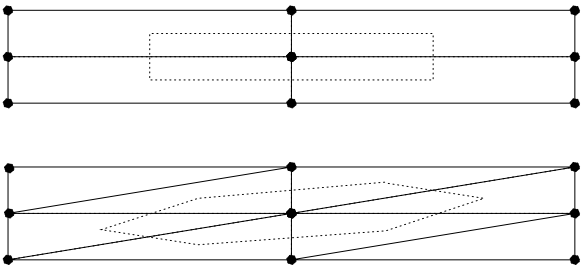

FIG. 2.2. *Median control-volumes for stretched quadrilateral and triangular elements*

For the convective terms, three different finite-volume discretizations have been implemented. The upwind scheme relies on a Roe Rieman solver [33] to compute a flux at the control-volume interfaces. Second-order accuracy is achieved by extrapolating vertex based variables to the control-volume interfaces based on gradient information evaluated by Green-Gauss contour integration around each control-volume boundary. Shock capturing is achieved using the smooth limiter of Venkatakrishnan [43]. This discretization is used exclusively in the preliminary study of sections 3 through 5.

The upwind scheme can be formulated as a central difference scheme with added dissipation, where the dissipation is constructed as a set of transformation matrices multiplied by the difference in left and right reconstructed states at a control-volume boundary. A matrix-based artificial dissipation scheme is obtained

by utilizing these same transformation matrices, but using them to multiply a difference of blended first and second differences rather than a difference of reconstructed states at control-volume boundaries. The traditional scalar artificial dissipation scheme is obtained by replacing the four eigenvalues u, u, $u+c$, $u-c$ in the transformation matrices of the matrix dissipation model by the maximum eigenvalue $|u|+c$, where u and c denote local fluid velocity and speed of sound, respectively. This scheme corresponds to the original scheme of Jameson et al. [8]. The scalar and matrix-based artificial dissipation schemes are used in sections 6 through 8.

The thin-layer form of the Navier-Stokes equations is employed in all cases, and the viscous terms are discretized to second-order accuracy by finite-difference approximation. For multigrid calculations, a first-order discretization is employed for the convective terms on the coarse grid levels for all cases.

The basic time-stepping scheme is a three-stage explicit multistage scheme with stage coefficients optimized for high frequency damping properties [41], and a CFL number of 1.8. Convergence is accelerated by a local block Jacobi preconditioner, which involves inverting a 4×4 matrix for each vertex at each stage [32, 22, 25, 26]. This approach, which can either be interpreted as a pre-conditioner, or as a local matrix time-step [11], has been shown to produce superior convergence rates for upwind schemes. No other techniques such as enthalpy damping or residual smoothing are employed [8].

The single equation turbulence model of Spalart and Allmaras [38] is utilized to account for turbulence effects. This equation is discretized and solved in a manner completely analogous to the flow equations, with the exception that the convective terms are only discretized to first-order accuracy.

3. Directional-Coarsening . In the context of unstructured meshes, there exists various strategies for implementing multigrid techniques. Two approaches that have been explored extensively by the author are the method of overset meshes, and the method of control-volume agglomeration [15, 37, 10, 18]. In the overset-mesh approach, a sequence of fine and coarse unstructured meshes is constructed either by hand, or in some automated fashion. These meshes are then employed in the multigrid algorithm, and variables are transferred between the various meshes of the sequence using linear interpolation. In the agglomeration approach, coarse levels are constructed by fusing together neighboring fine grid control volumes to form a smaller number of larger and more complex control volumes on the coarse grid.

While directional coarsening strategies can be employed in both multigrid approaches, for practical reasons we have chosen to utilize only the overset-mesh multigrid approach for these preliminary investigations. In fact, the same coarsening algorithm may be used for both approaches. In the overset-mesh approach, the graph coarsening algorithm is employed to select a subset of points from the fine grid from which the coarse grid will be formed. Once the coarse grid points have been determined, they must be triangulated in order to form a consistent coarse grid.

The coarsening algorithm is based on a weighted graph. Each edge of the mesh is assigned a weight which represents the degree of coupling in the discretization. In the true algebraic multigrid sense, these weights should be formed from the stencil coefficients. However, since the Navier-Stokes equations represent a system of equations, multiple coefficients exist for each edge. For simplicity, the edge weights are taken as the inverse of the edge length. For each fine grid vertex, the average and the maximum weight of all incident edges are precomputed and stored. This ratio of maximum to average weight is an indication of the local anisotropy in the mesh at each vertex. The coarsening algorithm begins by choosing an initial vertex as a coarse grid point or seed point, and attempts to remove neighboring points by examining the corresponding edge weights. If the ratio of maximum to average weights at the seed point is greater than α, (usually taken as $\alpha = 4$), then only the neighboring vertex along the edge of maximum weight is removed. Otherwise, (i.e.

in isotropic regions) all neighboring edges are removed. The next seed point is then taken from a priority list which contains points which are adjacent to points which have previously been deleted.

In the present implementation, the graph-based coarsening algorithm is only employed in the boundary-layer and wake regions. Once these regions have been coarsened, the remaining regions of the domain are regridded using a Delaunay advancing-front technique with user specified resolution. This approach is purely for convenience, since the original mesh is generated by a two-step procedure, which employs an advancing-layers technique in the regions of viscous flow, and an advancing-front Delaunay triangulation in regions of inviscid flow [16, 17]. The full weighted-graph coarsening algorithm will be implemented in the context of agglomeration multigrid in future work.

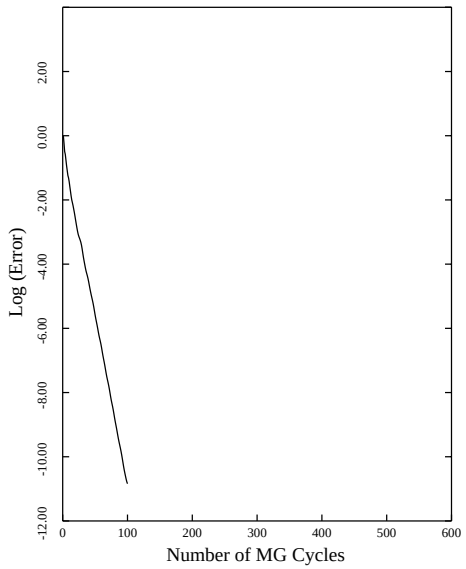

FIG. 3.1. *Multigrid Convergence Rate using Explicit Smoothing and Full-Coarsening for inviscid flow over NACA 0012 airfoil*

The first test case illustrates the convergence rates achievable for isotropic problems with the present algorithm. The inviscid transonic flow over a NACA 0012 airfoil is computed at a Mach number of 0.73 and incidence of 2.31 degrees. The mesh contains 5849 vertices and consists uniquely of isotropic triangular elements. The convergence history is documented in Figure 3.1. A total of 5 multigrid levels were employed, and a residual reduction of 11 orders of magnitude over 100 multigrid W-cycles was obtained. The overall convergence rate for this case is 0.77, which is very close to the theoretical limit of 0.75.

The second test case illustrates the stiffness induced by anisotropy. The viscous turbulent flow over the same geometry at the same conditions with a Reynolds number of 5 million is computed on the mesh depicted in Figure 2.1. This mesh contains a total of 4880 points. The cells on the airfoil surface have a height of 2.e-06 chords, and the maximum cell aspect ratio in the mesh is 20,000. This type of mesh is required in order to capture the boundary layer gradients. The computed Mach contours at these conditions are displayed in Figure 3.2. The convergence rate is depicted in Figure 3.3, using 5 multigrid levels which were constructed using the unweighted or full-coarsening version of the coarsening algorithm, as described in [18].

The slowdown in convergence over the inviscid test case is dramatic. After an initial phase of rapid convergence, the residual reduction rate slows down to less than 0.99 per multigrid W-cycle. Figure 3.3 also depicts the convergence rate of the same algorithm when a sequence of directionally coarsened grids is

employed in the multigrid algorithm. The improvement is substantial, yielding a residual reduction of 0.91 per multigrid V-cycle.

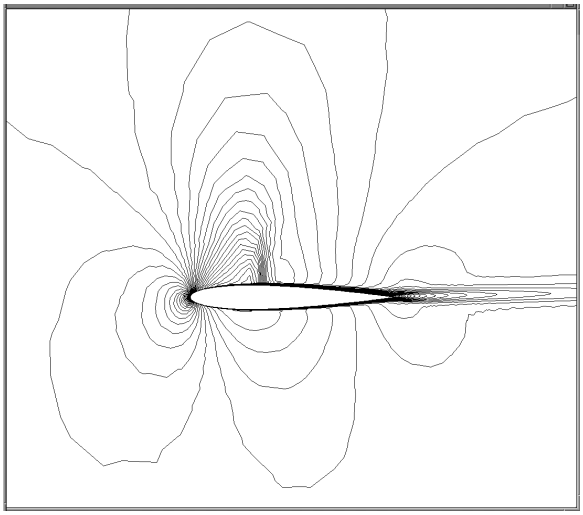

FIG. 3.2. *Computed Mach contours for viscous flow over NACA 0012 airfoil*

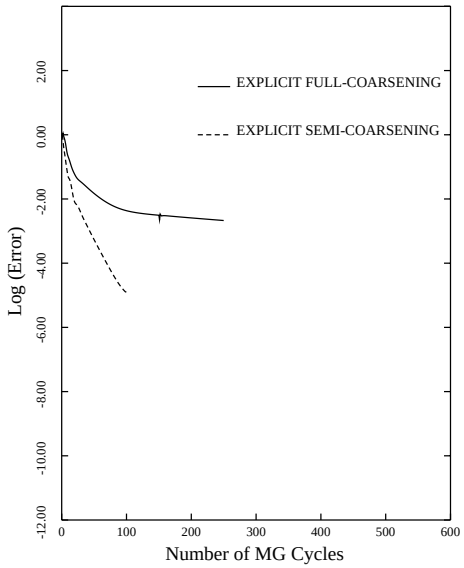

FIG. 3.3. *Comparison of Multigrid Convergence Rate using Explicit Smoothing and Full-Coarsening versus Explicit Smoothing and Semi-Coarsening for viscous flow over NACA 0012 airfoil*

4. Directional Implicit Smoothers . Although directional coarsening strategies for multigrid can achieve large increases in convergence speed, as demonstrated in the previous case, the coarse grids are more complex than those obtained in the full coarsening strategy. Note for example in the previous case that a V-cycle was required, since the W-cycle is impractical in this case (the amount of work involved is unbounded as the number of grid levels increases). As mentioned previously, the overhead required to store the coarse levels is also greatly increased in such cases.

An alternative approach is to use a directionally implicit smoother in conjunction with full coarsening multigrid. For structured grids, an example of a directionally implicit smoother is a line solver. Line solvers are attractive because they result in block-tridiagonal matrices which can be solved very efficiently. For unstructured grids, predetermined grid lines do not exist. However, line solvers can still be employed, provided lines are artificially constructed in the unstructured grid. Techniques for constructing lines in an unstructured grid have previously been described in the literature [7, 13]. In those efforts, lines which span the entire grid were constructed using unweighted graph techniques. In the present context, the role of the line solver is to relieve the stiffness induced by grid anisotropy. Therefore, lines are desirable only in regions of strong anisotropy, and in these regions they must propagate along the direction of strong coupling.

Given these requirements, an algorithm to build lines in an anisotropic mesh can be constructed using a weighted graph technique, in a manner analogous to the algorithm for directional coarsening described previously. The edge weights are defined as previously, and the ratio of maximum to average adjacent edge weight is pre-computed for every mesh vertex. The vertices are then sorted according to this ratio. The first vertex in this ordered list is then picked as the starting point for a line. The line is built by adding to the original vertex the neighboring vertex which is most strongly connected to the current vertex, provided

this vertex does not already belong to a line, and provided the ratio of maximum to minimum edge weights for the current vertex is greater than α, (using $\alpha = 4$ in all cases). The line terminates when no additional vertex can be found. If the originating vertex is not a boundary point then the procedure must be repeated beginning at the original vertex, and proceeding with the second strongest connection to this point. When the entire line is completed, a new line is initiated by proceeding to the next available vertex in the ordered list. Ordering of the initial vertex list in this manner ensures that lines originate in regions of maximum anisotropy, and terminate in isotropic regions of the mesh. The algorithm results in a set of lines of variable length. In isotropic regions, lines containing only one point are obtained, and the point-wise scheme is recovered.

On vector machines, the block-tridiagonal line solves must be vectorized across the lines. Because the lines have varying lengths, all lines must be made of similar length by padding the matrices of the shorter lines with zeros on the off-diagonals and ones on the diagonal entries, in such a way that zero additional corrections are generated at these locations by the implicit solver. To minimize the amount of padding required, the lines are sorted into groups, such that within each group, all lines are close in size to one another. Vectorization then takes place over the lines within each group. The size of the group also determines the amount of memory required for the line solves, since the tridiagonal matrices are constructed just prior to, and discarded just after the lines are solved, and all lines are uncoupled. For scalar machines, lines may be processed individually, and the memory requirements (i.e. additional working memory required by the implicit solver) are determined by the length of the longest line in the grid.

The implicit system generated by the set of lines can be viewed as a simplification of the general Jacobian obtained from a linearization of a backwards Euler time discretization, where the Jacobian is that obtained from a first-order discretization, assuming a constant Roe matrix in the linearization. For block-diagonal preconditioning, all off-diagonal block entries are deleted, while in the line-implicit method, the block entries corresponding to the edges which constitute the lines are preserved. The implicit line solver is applied as a preconditioner to the three-stage explicit scheme described previously. At each stage in the multi-stage scheme, the corrections previously obtained by multiplying the residual vector by the inverted block-diagonal matrix are replaced by corrections obtained by solving the implicit system of block-tridiagonal matrices generated from the set of lines. This implementation has the desirable feature that it reduces exactly to the block-diagonal preconditioned multi-stage scheme when the line length becomes one (i.e. 1 vertex and zero edges), as is the case in isotropic regions of the mesh.

As an example, the viscous flow case of the previous section has been recomputed using the line-implicit solver with full-coarsening multigrid. The set of lines generated in the mesh of Figure 2.1 are depicted in Figure 4.1. The lines extend through the boundary layer and wake regions, and have mostly a length of 1 (i.e. 1 vertex and zero edges) in the regions of inviscid flow where the mesh is isotropic. A total of 5 meshes was employed in the multigrid sequence. The convergence rate for this algorithm is depicted in Figure 4.2. The residuals are reduced by over 4 orders of magnitude in 100 cycles, which corresponds to a residual reduction rate of 0.92 per multigrid W-cycle. This rate is close to that obtained by the point-wise scheme using directional coarsening. However, the coarse grids are of lower complexity in this case and a W-cycle has been used.

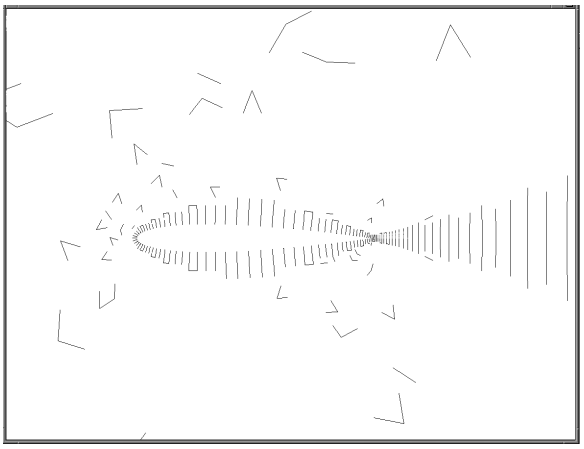

FIG. 4.1. *Implicit lines produced by the current algorithm on the grid of Figure 2.1*

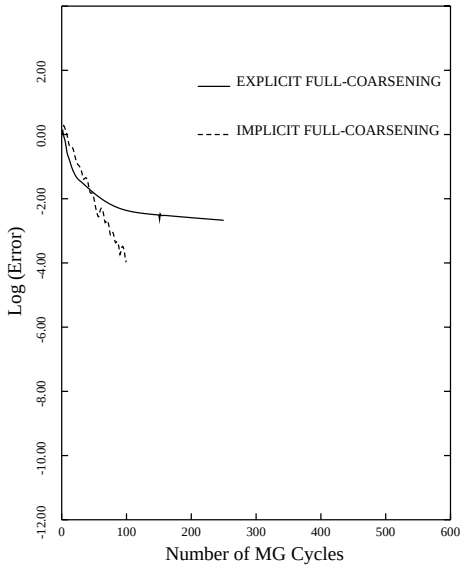

FIG. 4.2. *Comparison of Multigrid Convergence Rate using Explicit Smoothing and Full Coarsening versus Implicit Line Solver and Full-Coarsening for viscous flow over NACA 0012 airfoil*

5. Combining Directional Coarsening and Smoothing . There are obvious similarities between the directional coarsening algorithm and the technique used to construct lines for the directional implicit method. These two techniques can be combined, in a coupled manner, to produce a more robust and efficient overall algorithm. The simplest way to combine these techniques is to use the pre-conditioned line-implicit smoother with a sequence of directionally coarsened coarse multigrid levels. In order to more closely couple these two techniques, we use exactly the same criteria for coarsening and for line construction. This ensures that coarsening will proceed in the same direction and along the lines determined for the implicit solver.

An example of this combined algorithm is depicted by the convergence plot in Figure 5.1. In this case, the combined directional-implicit-coarsening algorithm has been used to solve the same viscous turbulent flow as described in the previous sections. The fine mesh for this case is similar to the one displayed in Figure 2.1, but contains 5828 points, and the mesh cells near the airfoil boundary have a height of 2.e-07 chord lengths, and the maximum aspect-ratio cell in the mesh is 200,000. This represents an order of magnitude more anisotropy than the previous mesh. While this type of normal boundary layer resolution is probably excessive for the present case, it is nonetheless representative of what is required for flight Reynolds number simulations of large aircraft. Even on this extremely stretched grid, the residuals are reduced by over 4 orders of magnitude over 100 cycles, which results in a average convergence rate of 0.92 per multigrid V-cycle. This rate is comparable to that achieved by either algorithm separately on the previous case. However, on this more highly stretched grid, neither algorithm alone could deliver this type of performance. Furthermore, the monotonic behavior of the current convergence history is a good indication of robustness.

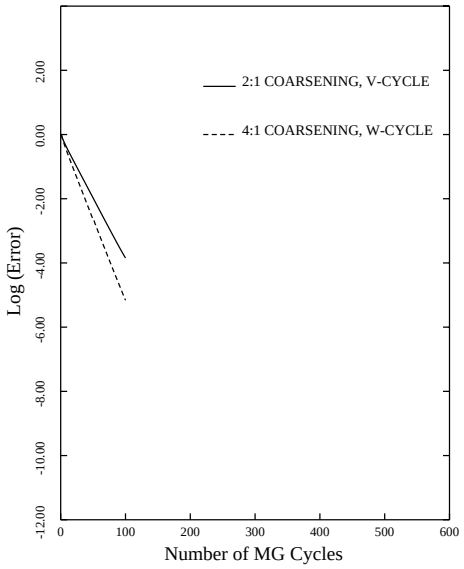

FIG. 5.1. *Multigrid Convergence Rate using Implicit Line-Solver and Semi-Coarsening for viscous flow over NACA 0012 airfoil*

On the other hand, this case is still plagued by the high coarse grid complexities of the semi-coarsening approach. However, these two techniques, directional coarsening and directional implicit smoothing, are two strategies for treating the same problem. In this respect they are more overlapping in nature than complementary, and one of these techniques may be relaxed somewhat. We therefore propose to perform directional-coarsening as described previously, along the direction of the implicit lines, but at a faster coarsening rate of 4:1. Therefore, rather than remove every second point along the implicit lines, we remove three points for every preserved coarse grid point along the implicit lines. In isotropic regions, the coarsening algorithm remains unchanged. This has the effect of generating a sequence of coarse grids which has roughly the same complexity as that obtained by the full-coarsening technique. To illustrate this approach, the same case has been recomputed using the line-implicit smoother and directional coarsening at a 4:1 rate. The convergence rate is compared with that obtained previously in Figure 5.1. The average residual reduction rate for this case is 0.88. The fact that this rate is even faster than that achieved in the previous example is attributed to the use of W-cycles in the current calculation, which is made possible due to the low complexity of the coarse grids.

6. Insensitivity to Aspect-Ratio and Discretization . While the above results are encouraging, they represent a single data point for a relatively easy problem on a relatively coarse grid. The goal of this section is to demonstrate the robustness of the present approach for more difficult and realistic problems, as well as the insensitivity of the approach to different discretization schemes and varying grid aspect-ratios.

The first test case consists of the computation of transonic flow over an RAE 2822 airfoil at a Mach number of 0.73, an incidence of 2.31 degrees, and a Reynolds number of 6.5 million. Three different grids were used for this computation. All three grids contain the same distribution of boundary points, but different resolutions in the direction normal to the boundary and wake regions. The first grid contains a normal wall spacing of 10^{-5} chords, and a total of 12,568 points, while the second grid contains a normal wall spacing

of 10^{-6} chords, and 16,167 points, and the third grid a normal wall spacing of 10^{-7} chords, and 19,784 points. The cells in the boundary layer and wake regions are generated using a geometric progression of 1.2 for all three grids. The second grid contains what is generally regarded as suitable normal and streamwise resolution for accurate computation of this type of problem, while the first and third grids are most likely under-resolved and over-resolved in the direction normal to the boundary layer, respectively. The second grid is displayed in Figure 6.1, while the Mach contours of the solution computed on this grid are displayed in Figure 6.2.

The solution algorithm is the same as described previously, namely directional coarsening multigrid combined with implicit line solvers. Coarsening in the boundary layer and wake regions occurs at a 4:1 rate, and the multigrid W-cycle is used exclusively. All results were obtained using the three-stage multi-stage scheme described previously. A five-stage scheme yielded faster convergence rates on a per cycle basis, but broke even on a cost (cpu time) basis. The matrix-based artificial dissipation discretization described in section 2 is used in this case. This scheme makes use of the same transformation matrices as the upwind scheme, but utilizes these to multiply a difference of second-differences rather than a difference of left and right reconstructed states. (First-order dissipation is omitted).

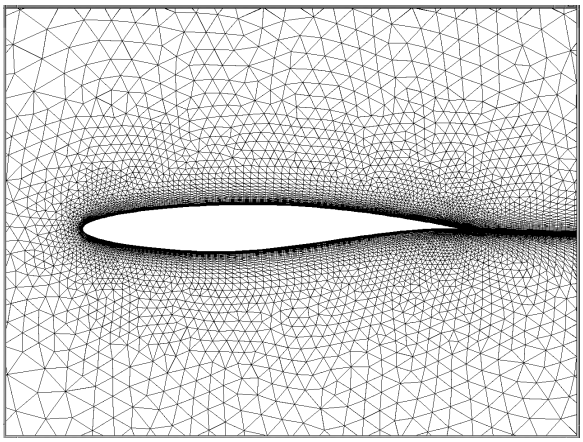

FIG. 6.1. *Unstructured Grid Used for Computation of Transonic Flow Over RAE 2822 Airfoil. Number of Points = 16167, Wall Resolution =* 10^{-6} *chords*

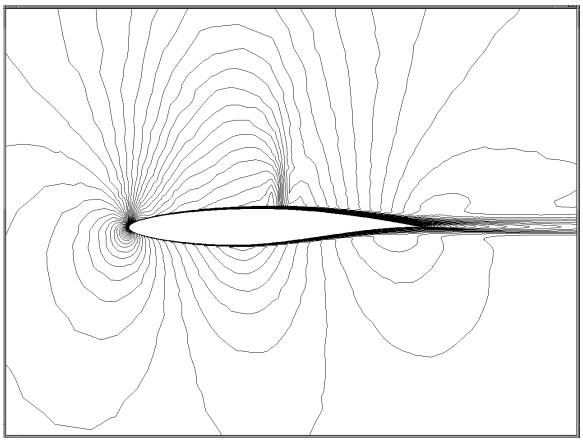

FIG. 6.2. *Computed Mach Contours on above Grid. Mach = 0.73, Incidence = 2.31 degrees, Re = 6.5 million*

Figure 6.3 depicts the observed convergence rates on all three grids for the matrix dissipation scheme. A twelve order reduction in the residuals is reached in all cases between 200 and 300 multigrid cycles. Given that these three grids represent a two-order-of-magnitude variation in grid aspect ratio, these convergence rates can be qualified as essentially insensitive to the degree of grid anisotropy.

Figure 6.4 depicts the convergence rates of similar computations using the scalar dissipation model. The scalar dissipation model, which corresponds to the original scheme of Jameson et al [8], is obtained by replacing the four eigenvalues $u, u, u+c, u-c$ in the transformation matrices of the matrix dissipation model by the maximum eigenvalue $|u|+c$. In this case, all matrices become diagonal and the dissipation is decoupled between the four equations of the system. The Jacobi preconditioning matrix also becomes diagonal, and the Jacobi preconditioner reduces to local scalar time-stepping [26]. The convergence rates in Figure 6.4 for this scheme are somewhat slower than those observed for the matrix dissipation model in Figure 6.3. However, they are relatively insensitive to the degree of grid stretching, and are much faster than the rates

achieved by the corresponding full-coarsening explicit multigrid algorithm (c.f. Figure 7.3). It is interesting to note that the convergence rate actually improves with increased grid anisotropy. This is perhaps due to the fact that the line solver more closely approximates a direct solver as the grid stretching increases. It should be noted that no averaging of the eigenvalues between the streamwise and normal directions has been performed in these cases. Such averaging techniques have typically been required to enhance convergence for scalar dissipation schemes on stretched grids [14, 28].

According to the analysis presented in [26], a scalar dissipation scheme of this type should be much more difficult to converge on highly stretched grids than a matrix-based dissipation scheme. Indeed, in a matrix or upwind dissipation discretization, only the acoustic modes are strongly coupled in the direction normal to the boundary layer when large grid stretching is present. This is due to the fact that the flow is mostly aligned with the grid in these regions, and thus the low normal dissipation due to flow alignment naturally counter-balances the stronger coupling due to the large grid aspect ratio, for the convective modes. The fact that the present algorithm is capable of efficiently converging the scalar-dissipation discretization and does so independently of the degree of grid stretching is a good indication of the robustness of the present approach. Furthermore, this property is important even for upwind and matrix dissipation discretizations in regions where the flow may not be aligned with highly stretched grid cells, such as in wake regions where the grid stretching does not accurately correspond to the physical wake direction.

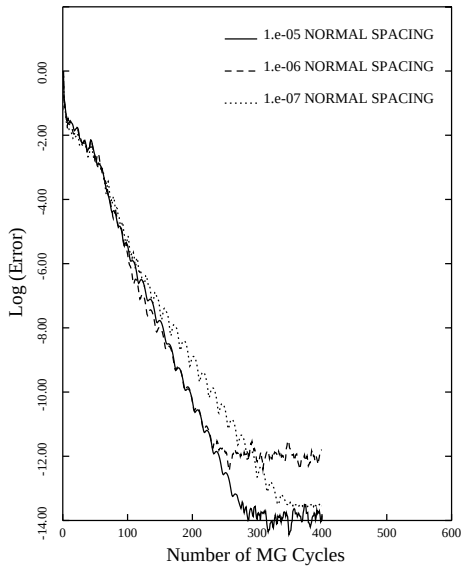

FIG. 6.3. *Comparison of Convergence Rates obtained on Grids of Differing Normal Resolution using Directional-Coarsening-Implicit Multigrid for Matrix Dissipation Scheme*

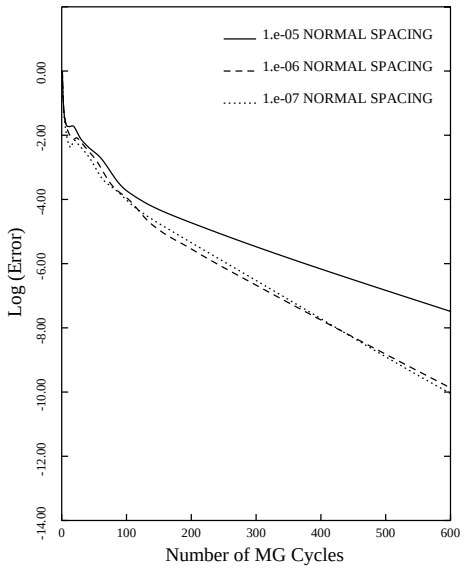

FIG. 6.4. *Comparison of Convergence Rates obtained on Grids of Differing Normal Resolution using Directional-Coarsening-Implicit Multigrid for Scalar Dissipation Scheme*

7. Relieving Other Types of Stiffness . While the results of the previous section illustrate the insensitivity of the present algorithm to grid aspect ratio and discretization models, the observed convergence rates remain substantially slower than those obtained for an isotropic inviscid problem, as in Figure 3.1, for example. Since the present scheme is insensitive to the grid aspect ratio, this slowdown is most likely due to other types of stiffness in the system of equations to be solved. For example, the disparity in eigenvalues caused by a very low Mach number flow cannot be adequately handled by the present approach. This type of

stiffness, which is inherent in the governing equations, rather than due to the grid topology, is often treated using preconditioning techniques [40, 11, 4, 44]. Conversely, it is now well known that physically based preconditioners can be used to construct more effective multigrid smoothers [12, 42].

In a first step towards further improving the convergence rates of the present scheme, a low Mach number preconditioner has been implemented in conjunction with the directional-coarsening-smoothing scheme. The particular implementation adopted follows the approach described in [27], which is based on the preconditioner of [4, 44]. This implementation is attractive because it does not require any change of variables in the current code. Traditionally, such preconditioners are described as a matrix multiplying an explicit updating scheme, and a similar matrix-based modification to the dissipation terms, which improves the accuracy at low Mach numbers. Thus, an original (first-order) matrix-based dissipation explicit scheme:

$$Vol_i.\frac{\Delta w_i}{\Delta t_i} = \sum_{k=1}^{neighbors} \frac{1}{2}(\mathbf{F}(w_i) + \mathbf{F}(w_k)).\mathbf{n}_{ik} - \frac{1}{2}|\mathbf{A}_{ik}|(w_k - w_i) \tag{7.1}$$

is replaced by:

$$\mathbf{P}^{-1}(Vol_i.\frac{\Delta w_i}{\Delta t_i}) = \sum_{k=1}^{neighbors} \frac{1}{2}(\mathbf{F}(w_i) + \mathbf{F}(w_k)).\mathbf{n}_{ik} - \frac{1}{2}\mathbf{P}^{-1}|\mathbf{P}\mathbf{A}_{ik}|(w_k - w_i) \tag{7.2}$$

where $\mathbf{P}$ is the preconditioning matrix, and $\mathbf{P}^{-1}|\mathbf{P}\mathbf{A}_{ik}|$ is the matrix modification to the dissipation terms. In the above equations, w_i represents the flow variables at vertex i and w_k represents the flow variables at neighboring vertices k of i. The convective fluxes are denoted by $\mathbf{F}(w)$, $\mathbf{n}_{ik}$ represents the normal vector of the control volume face separating the neighboring vertices i and k, and $\mathbf{A}_{ik}$ is the flux Jacobian evaluated in the direction normal to this face.

In the present work, we wish to implement this type of "preconditioner" in the context of a point-implicit (Jacobi-preconditioned) or line-implicit scheme. Since the low Mach number preconditioning matrix is a point-wise matrix, its implementation for point-implicit schemes is similar as for line-implicit, or any implicit scheme. The point-implicit scheme is derived by setting the right-hand-side on equation (7.1) equal to zero, and solving these equations with a Newton iteration, where only the (block-diagonal) entries corresponding to point i are retained in the Newton Jacobian. The flux $\mathbf{F}(w_i)$ does not contribute to this Jacobian, since it is integrated around the closed boundary of the control volume for point i. The resulting point-implicit scheme can be written as:

$$(\sum_{k=1}^{neighbors} \frac{1}{2}|\mathbf{A}_{ik}|)\Delta w_i = \sum_{k=1}^{neighbors} \frac{1}{2}(\mathbf{F}(w_i) + \mathbf{F}(w_k)).\mathbf{n}_{ik} - \frac{1}{2}|\mathbf{A}_{ik}|(w_k - w_i) \tag{7.3}$$

In the scalar dissipation case, the matrix $\mathbf{A}_{ik}$ becomes diagonal, and the summation multiplying the correction on the left-hand side represents the usual time-step estimate:

$$\frac{Vol_i}{\Delta t_i} = (\sum_{k=1}^{neighbors} \frac{1}{2}|\mathbf{A}_{ik}|) \tag{7.4}$$

thus verifying that this scheme reduces to local time-stepping for scalar dissipation discretizations. Taking the point-wise linearization of the modified discretization (i.e. right-hand-side of equation (7.2)) leads to the scheme:

$$(\sum_{k=1}^{neighbors} \frac{1}{2}\mathbf{P}^{-1}|\mathbf{P}\mathbf{A}_{ik}|)\Delta w_i = \sum_{k=1}^{neighbors} \frac{1}{2}(\mathbf{F}(w_i) + \mathbf{F}(w_k)).\mathbf{n}_{ik} - \frac{1}{2}\mathbf{P}^{-1}|\mathbf{P}\mathbf{A}_{ik}|(w_k - w_i) \tag{7.5}$$

Using equation (7.4), for the scalar dissipation case, the scheme described by equation (7.5) is equivalent to that described by equation (7.2) provided one uses a locally averaged value of the preconditioning matrix, and the appropriate eigenvalues of the modified system are used in the time-step estimate. Therefore, low Mach number preconditioning may be implemented by simply modifying the dissipation matrices as per equation (7.2), and taking this modification into account in the point-wise linearization that is required for the Jacobi scheme or preconditioner. The low Mach number "preconditioning" matrix arises naturally from this linearization. This offers a different view-point of the mechanism involved for relieving low Mach number stiffness in compressible flow formulations. Rather than a technique which attempts to alter the time history of the convergence process, this type of "preconditioning" may be thought of as one which simply alters the discretization of an implicit scheme to yield a more consistent and less stiff system of equations.

Figure 7.1 depicts the convergence rates achieved by the directional-coarsening-implicit multigrid scheme combined with low Mach number preconditioning for the matrix dissipation discretization for flows at various Mach numbers over an RAE 2822 airfoil on the grid of Figure 6.1 (normal wall resolution of 10^{-6} chords). While the un-preconditioned cases slow down dramatically with decreasing freestream Mach number, the preconditioned cases produce convergence rates which are virtually identical over a two-order-of-magnitude variation in the freestream Mach number. In order to prevent singularities at stagnation points, the specification of a cutoff Mach number is required [44, 27], below which preconditioning is omitted. This is generally specified as a function of the freestream Mach number. In this study, the required cutoff value was such that no preconditioning could be applied in the transonic case.

Figure 7.2 illustrates the same convergence histories for the scalar dissipation scheme. The non-preconditioned convergence rates for all Mach numbers are slower for the scalar dissipation scheme than for the matrix dissipation scheme, as expected. However, the preconditioned convergence rates for the two schemes are almost identical. These convergence rates are also relatively insensitive to the freestream Mach number. Even the transonic case benefits from preconditioning when scalar dissipation is employed. This is attributed to the use of a lower cutoff Mach number which was possible only in the scalar dissipation scheme.

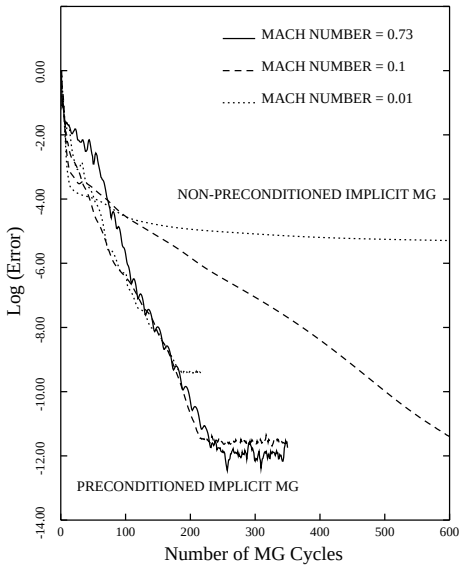

FIG. 7.1. *Comparison of Convergence Rates obtained on Grid of Figure 6.1 for Various Freestream Mach numbers with and without Low-Mach-number Preconditioning for Matrix Dissipation Scheme. Note that Mach = 0.73 case could not be run with Low-Mach-number Preconditioning*

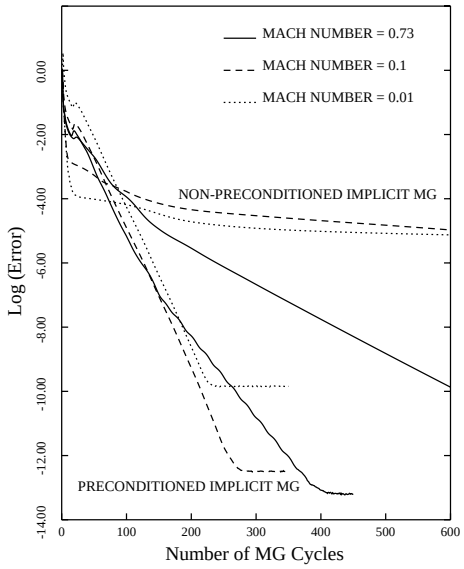

FIG. 7.2. *Comparison of Convergence Rates obtained on Grid of Figure 6.1 for Various Freestream Mach numbers with and without Low-Mach-number Preconditioning for Scalar Dissipation Scheme*

In Figure 7.3, we return to the transonic case, and compare the pre-conditioned directional-coarsening-implicit multigrid scheme with the full coarsening explicit multigrid scheme for the scalar dissipation discretization. This figure contrasts the decaying convergence rates of the original algorithm as the grid stretching increases, with the insensitivity of the improved algorithm to grid stretching. The original algorithm employs a five stage scheme with residual smoothing, while the current algorithm employs a three-stage scheme with no residual smoothing but with directional line solves. Although the current algorithm is a factor of 4 more expensive than the original algorithm, on a per cycle basis, this is mainly due to incomplete optimization and higher coarse grid complexities in the inviscid regions of flow. It is expected that the current algorithm can be made to be approximately equal to the original algorithm on a per-cycle basis through additional optimization and reduction in coarse grid complexity.

The current algorithm has been employed to compute the flow over a three-element airfoil geometry. The grid employed for this computation is shown in Figure 7.4. The grid contains a total of 30562 points, and a normal spacing of 10^{-5} chords at the wall and in the near wake regions. The implicit lines extracted from this grid by the algorithm are depicted in Figure 7.5. They occur mainly in the boundary-layer and wake regions.

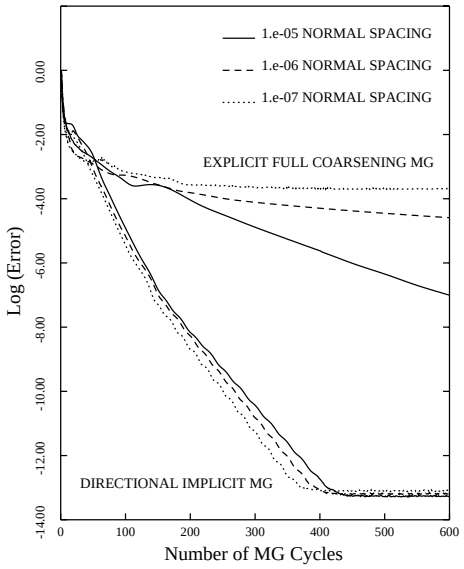

FIG. 7.3. *Demonstration of Aspect-Ratio Independent Convergence Rates Achieved by the Preconditioned-Implicit Multigrid Scheme and Comparison with Aspect-Ratio Sensitive Convergence Rates of Regular Multigrid Approach for Mach = 0.73 case (scalar dissipation scheme)*

A qualitative view of the computed Mach contours is depicted in Figure 7.6. The freestream Mach number is 0.2, and the incidence and Reynolds number are 16 degrees and 5 million, respectively. The convergence rates obtained for the scalar dissipation scheme on this case are depicted in Figure 7.7. The original full-coarsening explicit multigrid algorithm converges very slowly in the asymptotic range. The directional-coarsening-implicit scheme delivers substantially faster convergence, but is still much slower than that observed for the transonic cases. The low-Mach number preconditioned version of this latter scheme provides substantially faster convergence, yielding an asymptotic convergence rate of 0.955 prior to the slowdown after 300 cycles, using a three-stage smoother. This rate is still somewhat slower than that achieved by the same algorithm for the single airfoil cases. This is attributed to the use of a global cutoff Mach number in the low-Mach number preconditioner [44, 27], which is required to prevent singularities in regions of stagnation flow. Inspection of the flow in the three-element airfoil case reveals much larger regions of very low Mach number flow as compared to the single airfoil cases (for the same freestream Mach number). The use of a global cutoff based on the freestream Mach number is obviously inconsistent with the notion of a *local* preconditioner. However, as documented in the literature, it has been found that this type of cutoff is required for robustness. Further research is required to resolve this problem.

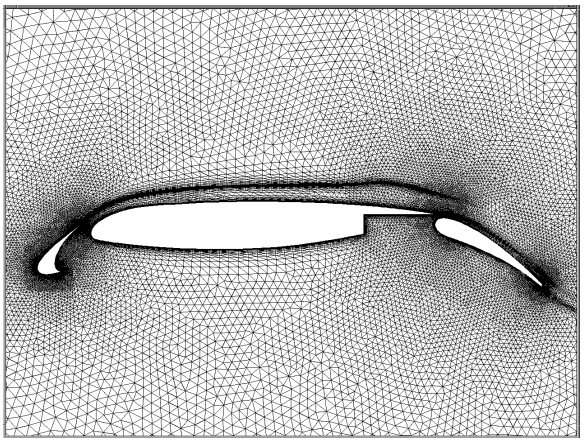

FIG. 7.4. *Unstructured Grid Used for Computation of Subsonic Flow Over Three-Element Airfoil Geometry. Number of Points = 30562, Wall Resolution =* 10^{-5} *chords*

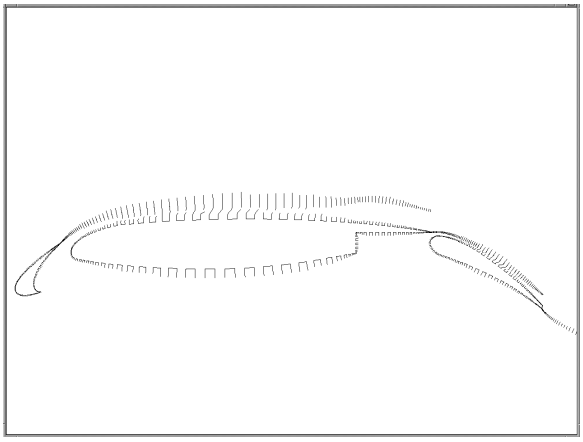

FIG. 7.5. *Implicit Lines Constructed on Grid about Three-Element Airfoil*

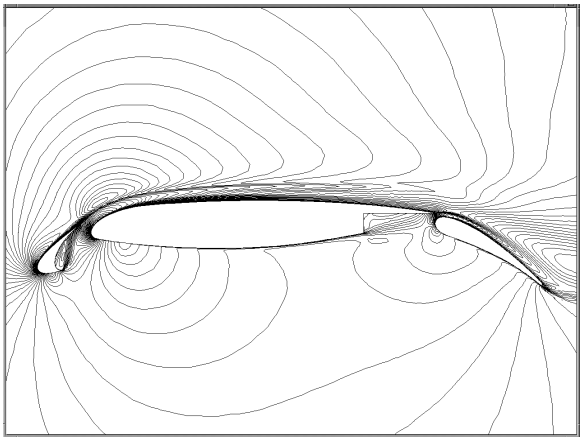

FIG. 7.6. *Computed Mach Contours for Three-Element Airfoil Calculation. Mach = 0.2, Incidence = 16 degrees, Re number = 5 million*

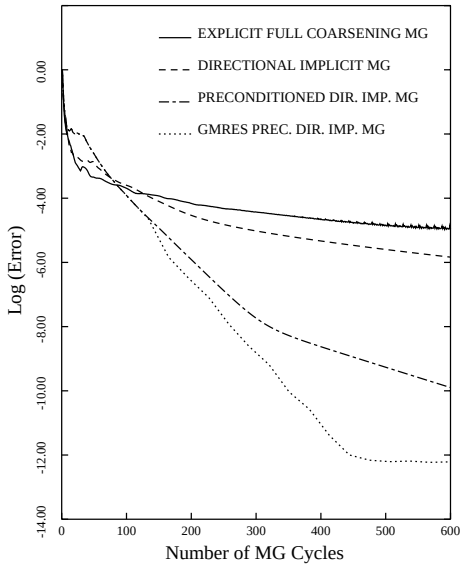

FIG. 7.7. *Comparison of Convergence Rates obtained for Flow Over Three-Element Airfoil Using Various Preconditioned Multigrid Schemes for Scalar Dissipation Discretization*

8. Incorporating a Krylov Acceleration Technique:Preconditioning [3] **.** Further gains in efficiency can be obtained by incorporating a Krylov acceleration technique, such as GMRES [35]. In the present context, the low-Mach number preconditioned directional-coarsening multigrid scheme, using the line-implicit smoother (which itself may be viewed as a preconditioner) is used as a preconditioner to GMRES, thus the term *Preconditioning*[3]. This approach has been demonstrated previously for multigrid-based

solvers in the literature [45, 25]. GMRES can be formulated to solve a set of non-linear equations in a matrix-free manner by computing Jacobian-vector products by finite differencing the residual vector. This approach is particularly attractive since it may be implemented as an addition to an existing solver with minimal modifications to the existing code.

The addition of GMRES incurs little additional cpu time, measured on a multigrid cycle basis, but requires considerable additional memory, since a solution vector must be stored for each of the Krylov search directions. Typically, 20 to 30 search directions are employed. The GMRES subroutine employed in this work is the one developed in [45].

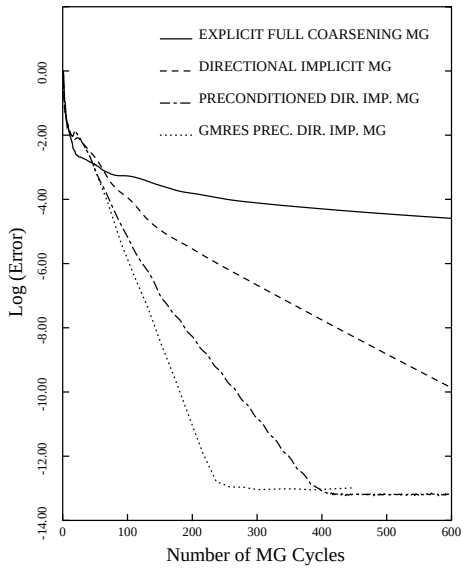

FIG. 8.1. *Comparison of Convergence Rates obtained on Grid of Figure 6.1 using Various Multigrid Schemes for Transonic Case using Scalar Dissipation*

Figure 8.1 depicts the convergence rate obtained using the preconditioned GMRES technique with 20 search directions for the transonic flow over an RAE 2822 airfoil using the scalar dissipation discretization, and compares it with several variants of the algorithms discussed in the preceding sections. As can be seen, the addition of the GMRES routine to the low-Mach number preconditioned directional-coarsening line-implicit scheme nearly doubles the convergence rate as compared to the same scheme without GMRES. The asymptotic convergence rate of this scheme is 0.87 per multigrid cycle, which is only a factor of 2 away from the optimum goal of 0.75.

Figure 7.7 depicts the same schemes applied to the three-element airfoil case. In this case, the addition of GMRES using 30 search directions produces a smaller gain in speed of convergence, at least prior to the slowdown exhibited by the best non-GMRES scheme after 8 orders of magnitude residual reduction. It is of academic interest that GMRES is successful in eliminating this slowdown, since engineering computations would not normally be carried out this far. The best asymptotic convergence rate achieved is roughly 0.945, which is approximately twice as slow as that achieved for the single airfoil transonic case. This is thought to be due to the presence of larger regions of lower velocity fluid in the domain. Figures 7.7 and 8.1 summarize the relative gains achieved by the various schemes and their compounding effects for these two cases.

9. Further Work . In the above discussion, the comparisons between various schemes are made on a per cycle basis. While this is useful for determining the relative effectiveness of each technique as a multigrid smoother, and the degree to which the overall algorithm approaches the hypothetical "optimal" algorithm, it does not convey the relative costs of these various schemes. One of the reasons direct cpu comparisons have not been made is that the current code is not sufficiently optimized to provide a fair comparison with the baseline algorithm. Another reason, is that the technique used for coarse grid construction results in less than optimal coarse grid complexity since the inviscid regions of the flow are actually regridded, rather than coarsened by point removal. (However, the boundary layer and wake regions exhibit optimal coarsening). On the other hand, given the magnitude of the gains in speed of convergence, it should be evident that even the currently non-optimized implementation is much more efficient than the original scheme. While the current algorithm is roughly a factor of 4 more costly on a per cycle basis than the original explicit multigrid algorithm with residual smoothing, it is anticipated that this cost can be reduced to be comparable to that of the explicit multigrid algorithm.

The ultimate goal of this work is to incorporate these techniques into the more practical agglomeration or algebraic multigrid method described in [18, 20]. The development of an optimal grid coarsening scheme for agglomeration multigrid is currently under development. Figure 9.1 illustrates the first agglomerated level of a stretched unstructured grid, where the agglomeration has been constrained to proceed along the implicit-lines in the boundary layer regions, at a rate of 4:1. In the isotropic regions of the mesh, this algorithm reverts to that developed in [18, 20].

The use of line solvers can lead to additional complications for distributed memory parallel implementations. Since the classical tridiagonal line solve is an inherently sequential operation, any line which is split between multiple processors will result in processors remaining idle while the off-processor portion of their line is computed on a neighboring processor. However, the topology of the line sets in the unstructured grid is such that it should be possible to partition the mesh in such a manner that lines are completely contained within an individual processor, with minimal penalty (in terms of processor imbalance or additional numbers of cut edges).

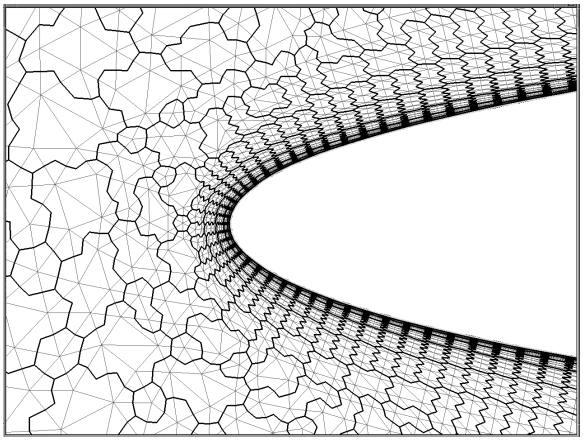

FIG. 9.1. *Example of Agglomerated Grid Using 4:1 Coarsening along Lines in Boundary-Layer Region*

This can be achieved by using a graph-based mesh partitioner with weights assigned to each edge. High edge weights correspond to edges which should not be cut between partitions. Figure 9.2 illustrates a 32-way

partition on the unstructured grid of Figure 7.4 , using the partitioner described in [9], both with uniform weights, and with large weights assigned to the edges which constitute the implicit-lines for the flow solver (c.f. Figure 7.5). While the un-weighted partition results in 269 cut lines, the weighted partition results in no cuts across lines with a 8.5 % increase in total cut edges.

Although the proposed algorithms in this work have demonstrated convergence rates independent of the degree of grid stretching, the convergence rates obtained for viscous flows are still a factor of 2 to 4 slower than what may be achieved for simple inviscid flow problems. Additional research is required to further reduce these rates consistently for all types of viscous flow problems. Perhaps the most promising avenue of research is to develop more sophisticated preconditioners in an effort to provide a better multigrid smoother at low additional cost [12, 42]. Of immediate concern is the need to formulate a more consistent cutoff criterion for low-Mach number preconditioners [39, 5].

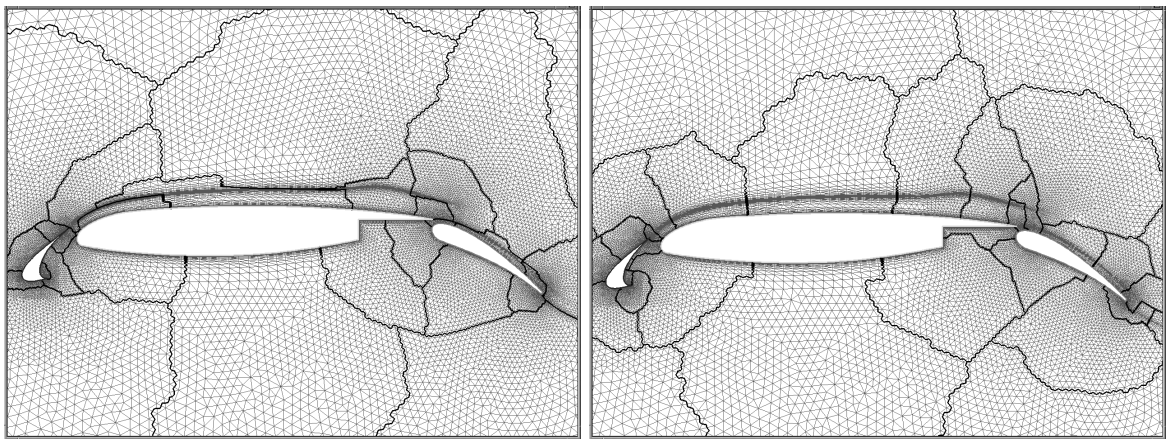

FIG. 9.2. *Comparison of Unweighted (Left) and Weighted (Right) 32-Way Partition of Unstructured Mesh for Parallel Computation*

10. Acknowledgments. The author would like to thank Stephen Guattery of ICASE and Alex Pothen of Old Dominion University for providing the un-weighted and weighted graph partitions in Figure 9.2.

REFERENCES

[1] S. R. ALLMARAS, *Analysis of semi-implicit preconditioners for multigrid solution of the 2d compressible navier-stokes equations*, in Proceedings of the 12th AIAA CFD Conference, San Diego CA, June 1995. AIAA Paper 95-1651-CP.

[2] T. J. BARTH AND S. W. LINTON, *An unstructured mesh newton solver for compressible fluid flow and its parallel implementation.* AIAA paper 95-0221, Jan. 1995.

[3] A. BRANDT, *Multigrid techniques with applications to fluid dynamics:1984 guide*, in VKI Lecture Series, Mar. 1984, pp. 1–176.

[4] Y. CHOI AND C. MERKLE, *The application of preconditioning to viscous flows*, J. Comp. Phys., 105 (1993).

[5] D. L. DARMOFAL AND P. J. SCHMID, *The importance of eigenvectors for local preconditioners of the euler equations*, in Proceedings of the 12th AIAA CFD Conference, San Diego, CA, June 1995, pp. 102–117. AIAA Paper 95-1655-CP.

[6] J. Francescatto, *Résolution de l'équation de Poisson sur des maillages étirés par une methode multigrille*. INRIA Report No. 2712, Nov. 1995.

[7] O. Hassan, K. Morgan, and J. Peraire, *An adaptive implicit/explicit finite element scheme for compressible high speed flows*. AIAA Paper 89-0363, Jan. 1989.

[8] A. Jameson, W. Schmidt, and E. Turkel, *Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes*. AIAA Paper 81-1259, 1981.

[9] G. Karypis and V. Kumar, *A fast and high quality multilevel scheme for partitioning irregular graphs*, Tech. Report Technical Report 95-035, University of Minnesota, 1995. A short version appears in Intl. Conf. on Parallel Processing 1995.

[10] M. Lallemand, H. Steve, and A. Dervieux, *Unstructured multigridding by volume agglomeration: Current status*, Computers and Fluids, 21 (1992), pp. 397–433.

[11] B. v. W. T. Lee and P. Roe, *Characteristic time-stepping or local preconditioning of the Euler equations*, in Proceedings of the 10th AIAA CFD Conference, Honolulu, Hawaii, June 1991, pp. 260–282. AIAA Paper 91-1552-CP.

[12] J. F. Lynn and B. van Leer, *A semi-coarsened multigrid solver for the euler and navier-stokes equations with local preconditioning*, in Proceedings of the 12th AIAA CFD Conference, San Diego, CA, June 1995, pp. 242–252. AIAA Paper 95-1667-CP.

[13] D. Martin and Löhner, *An implicit linelet-based solver for incompressible flows*. AIAA Paper 92-0668, Jan. 1992.

[14] L. Martinelli and A. Jameson, *Validation of a multigrid method for the Reynolds-averaged Navier-Stokes Equations*. AIAA Paper 88-0414, Jan. 1988.

[15] D. J. Mavriplis, *Three-dimensional multigrid for the Euler equations*, AIAA Journal, 30 (1992), pp. 1753–1761.

[16] ——, *An advancing-front Delaunay triangulation algorithm designed for robustness*. AIAA paper 93-0671, Jan. 1993.

[17] ——, *Unstructured mesh generation and adaptivity*, in 26th VKI Lecture Series on Computational Fluid Dynamics, Lecture Series 1995-02, Mar. 1995.

[18] D. J. Mavriplis and V. Venkatakrishnan, *Agglomeration multigrid for two dimensional viscous flows*, Computers and Fluids, 24 (1995), pp. 553–570.

[19] ——, *A unified multigrid solver for the Navier-Stokes equations on mixed element meshes*. AIAA Paper 95-1666, June 1995.

[20] ——, *A 3D agglomeration multigrid solver for the Reynolds-averaged Navier-Stokes equations on unstructured meshes*, International Journal for Numerical Methods in Fluids, 23 (1996), pp. 527–544.

[21] E. M. D. J. Mavriplis and V. Venkatakrishnan, *Coarsening strategies for unstructured multigrid techniques with application to anisotropic problems*, in 7th Copper Mountain Conf. on Multigrid Methods, Apr. 1995, pp. 591–606. NASA Conference Publication 3339.

[22] E. Morano and A. Dervieux, *Looking for O(N) Navier-Stokes solutions on non-structured meshes*, in 6th Copper Mountain Conf. on Multigrid Methods, 1993, pp. 449–464. NASA Conference Publication 3224.

[23] W. A. Mulder, *A new multigrid approach to convection problems*, Journal of Computational Physics, 83 (1989), pp. 303–323.

[24] ——, *A high resolution Euler solver based on multigrid semi-coarsening and defect correction*, Journal of Computational Physics, 100 (1992), pp. 91–104.

[25] C. Ollivier-Gooch, *Towards problem-independent multigrid cnvergence rates for unstructured mesh methods i: Inviscid and laminar flows*, in Proceedings of the 6th International Symposium on CFD, Lake Tahoe, NV, Sept. 1995.

[26] N. Pierce and M. Giles, *Preconditioning on stretched meshes*. AIAA paper 96-0889, Jan. 1996.

[27] D. J. T. Pulliam and P. Buning, *Recent enhancements to OVERFLOW*. AIAA Paper 97-0644, Jan. 1997.

[28] R. Radespiel and R. C. Swanson, *An investigation of cell-centered and cell-vertex multigrid schemes for the Navier-Stokes equations*. AIAA paper 89-053, Jan. 1989.

[29] ———, *Progress with multigrid schemes for hypersonic flow problems*, Journal of Computational Physics, 116 (1995), pp. 103–122.

[30] W. K. A. R. Rausch and D. Bonhaus, *Implicit multigrid algorithms for incompressible turbulent flows on unstructured grids*, in Proceedings of the 12th AIAA CFD Conference, San Diego CA, June 1995. AIAA Paper 95-1740-CP.

[31] M. Raw, *Robustness of coupled algebraic multigrid for the Navier-Stokes equations*. AIAA paper 96-0297, Jan. 1996.

[32] K. Riemslagh and E. Dick, *A multigrid method for steady Euler equations on unstructured adaptive grids*, in 6th Copper Mountain Conf. on Multigrid Methods, NASA conference publication 3224, 1993, pp. 527–542.

[33] P. L. Roe, *Approximate Riemann solvers, parameter vectors and difference schemes*, J. Comp. Phys., 43 (1981), pp. 357–372.

[34] J. W. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid Methods, S. F. McCormick, ed., SIAM Frontiers in Applied Mathematics, Philadelphia, 1987, SIAM, pp. 73–131.

[35] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.

[36] D. Sidilkover, *A genuinely multidimensional upwind scheme and efficient multigrid solver for the compressible Euler equations*. ICASE Report 94-84, Oct. 1994.

[37] W. A. Smith, *Multigrid solution of transonic flow on unstructured grids*, in Recent Advances and Applications in Computational Fluid Dynamics, Nov. 1990. Proceedings of the ASME Winter Annual Meeting, Ed. O. Baysal.

[38] P. R. Spalart and S. R. Allmaras, *A one-equation turbulence model for aerodynamic flows*, La Recherche Aérospatiale, 1 (1994), pp. 5–21.

[39] B. v. E. T. C. H. Tai and L. Mesaros, *Local preconditioning in a stagnation point*, in Proceedings of the 12th AIAA CFD Conference, San Diego, CA, June 1995, pp. 88–101. AIAA Paper 95-1654-CP.

[40] E. Turkel, *Preconditioning methods for solving the incompressible and low speed compressible equations*, J. Comp. Phys., 72 (1987), pp. 277–298.

[41] B. van Leer, C. H. Tai, and K. G. Powell, *Design of optimally-smoothing multi-stage schemes for the Euler equations*. AIAA Paper 89-1933, June 1989.

[42] E. T. V. N. Vatsa and R. Radespiel, *Preconditioning methods for low speed flow*. AIAA Paper 96-2460, 1996.

[43] V. Venkatakrishnan, *On the accuracy of limiters and convergence to steady state solutions*, J. Comp. Phys., 118 (1995), pp. 120–130.

[44] J. M. Weiss and W. A. Smith, *Preconditioning applied to variable and constant density time-accurate flows on unstructured meshes*. AIAA Paper 94-2209, June 1994.

[45] L. B. W. N. J. Yu and D. P. Young, *GMRES acceleration of computational fluid dynamic codes*, in Proceedings of the 7th AIAA CFD Conference, July 1985, pp. 67–74. AIAA Paper 85-1494-CP.